Chapter 20
Stochastic Optimization

20.1 Introduction

Suppose $x \in \mathbb{S}$ is some vector in an n-dimensional search space $\mathbb{S}$ and let $\mathbb{H} : \mathbb{S} \to \mathbb{R}$ be a mapping from the search space $\mathbb{S}$ onto the real axis $\mathbb{R}$. The function $\mathbb{H}$ plays a particular role and is usually referred to as the *cost function*. A minimization problem can be defined in a very compact form:

> Find $x_0 \in \mathbb{S}$, such that $\mathbb{H}(x_0)$ is the *global* minimum of the cost function $\mathbb{H}$.

In analogue, a maximization problem with cost function $\mathbb{H}$ defines a minimization problem with cost function $\mathbb{G} = -\mathbb{H}$. The class of both problems is referred to as the class of optimization problems [1–3] and only minimization problems will be discussed here.

The reader might be aware that there are numerous applications in physics and related sciences. We list a few in order to remind ourselves of their fundamental importance:

- The set of linear equations $Ax = b$ is often regarded as the minimization problem: $\mathbb{H}(x) = \|Ax - b\|^2$ which can be beneficial for high dimensional problems.
- The quantum mechanical ground state energy E_0 is given by

$$E_0 = \min_{\Psi} \frac{\langle \Psi \,|\, H \,|\, \Psi \rangle}{\langle \Psi \,|\, \Psi \rangle} \,, \tag{20.1}$$

  where $|\Psi\rangle$ denotes the wave function and H is the Hamiltonian of the system.
- High dimensional and highly non-linear least squares fits. (More details can be found in Appendix H.)
- The equilibrium crystal structure of solids is obtained by minimization of the free energy.
- Protein folding is described by minimization of the forces in a *molecular dynamics* problem.

B.A. Stickler, E. Schachinger, *Basic Concepts in Computational Physics*,
DOI 10.1007/978-3-319-27265-8_20

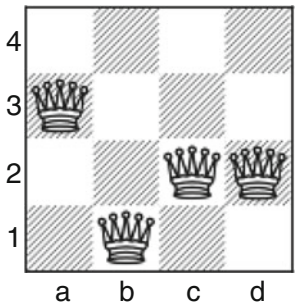
Whenever the cost function is at least once differentiable, methods of deterministic optimization can be applied [4]. (Two simple deterministic optimization methods are presented in Appendix I.) On the other hand, if $\mathbb{H}$ is not differentiable or too complex, due to a huge search space $\mathbb{S}$ or many local minima, methods of stochastic optimization [5] can be employed. The term stochastic optimization is used for methods which contain at least one step which is based on random number generation. Let us briefly give some examples of problems for which deterministic methods fail:

- The *Traveling Salesperson Problem* [6, 7]: A traveling salesperson has to visit L cities in a tour as short as possible under the constraint that he/she has to return to the starting point in the end. Each city has to be visited only once, hence the cities have to be ordered in such a way that the travel length becomes a global minimum. In particular, the cost function

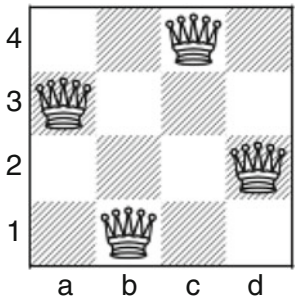
$$\mathbb{H}(\{i\}) = \sum_{\ell=1}^{L} |x_{i_{\ell+1}} - x_{i_\ell}| \ , \tag{20.2}$$

  has to be minimized. Here $\{i\}$ denotes a certain configuration of cities and we set $i_{L+1} = i_1$. Obviously, we cannot calculate the first derivative of $\mathbb{H}$ with respect to $\{i\}$, set it zero, and solve the problem in the classical way. On the other hand, a brute force approach of calculating $\mathbb{H}(\{i\})$ for all possible arrangements $\{i\}$ is not possible since we have $L!$ different possible routes. Since for one particular choice all L starting points and both travel directions yield the same result, we have to calculate $L!/(2L) = (L-1)!/2$ different configurations $\{i\}$. We would have about 10^{155} different choices for $L = 100$ cities! This clearly makes such an approach intractable.
- The arrangement of timetables under certain constraints. In particular, the design of timetables in schools, universities or at airports. This problem is also referred to as the *Nurse Scheduling Problem* [8].
- The ISING *spin glass* [9]: In contrast to the classical ISING model, the ISING spin glass is characterized by nearest neighbor interactions J_{ij} which are, in the most simple case, chosen to be $J_{ij} = +1$ and $J_{ij} = -1$ with the same probability. In this case the ground state below the critical temperature is not simply given by a configuration in which all spins point in the same direction. Of course, the ground state configuration in such a case can be highly degenerate. The fact that such a model can be simulated using MARKOV-chain Monte Carlo methods as they have been discussed within Chaps. 15 and 18 gives us some idea of how one may employ stochastic methods to solve optimization problems.
- The *N-Queens Problem* [10]: Place N queens on a $N \times N$ chessboard in such a way that no two queens attack each other. In particular, this means that two queens are not allowed to share the same row, the same column, and the same diagonal. It can be shown that the problem possesses solutions for $N \geq 4$. One defines a function $\mathbb{H}(\{n\})$ which counts the number of attacks in a certain configuration

$\{n\}$. For instance, for $N = 4$, the configuration



has $\mathbb{H}(\{n\}) = 2$. On the contrary, the configuration



solves the 4-queens problem and $\mathbb{H}(\{n\}) = 0$.

We concentrate here on some of the most basic methods of stochastic optimization: the method of *hill climbing*, the method of *simulated annealing*, and *genetic algorithms*. Ideas on which several more advanced techniques are based will be sketched in Sect. 20.5.

20.2 Hill Climbing

The method of *hill climbing* [11] is probably one of the most simple methods of stochastic optimization. Given a cost function $\mathbb{H}(x)$, we execute the following steps:

1. Choose an initial position x_0.
2. Randomly pick a new x_n from the *neighborhood* of x_{n-1}.
3. Keep x_n if $\mathbb{H}(x_n) \leq \mathbb{H}(x_{n-1})$.
4. Terminate the search if no new x_n can be found in the neighborhood of x_{n-1}.

We note that the algorithm requires a *neighborhood* relation. This relation is to be defined for each particular problem. For instance, in the case of the traveling salesperson problem it is by no means clear what a configuration in the *neighborhood* of a certain route $\{i\}$ should mean. To elaborate on this problem we concentrate here on two particular problems which help to demonstrate how such a neighborhood relation can be defined.

In the traveling salesperson problem or in the ISING spin glass model the neighborhood of a route $\{i\}$ or of a configuration $\mathscr{C}$ can be defined as the set of all routes $\{i\}$ in which two cities have been interchanged or as the set of all configurations $\mathscr{C}$ in which one spin has been flipped.

On the other hand, if the search space $\mathbb{S} = \mathbb{R}^n$ we may define the neighborhood as the number of points within an n-sphere of radius r centered at $z \equiv x_{n-1}$. It is rather simple to sample points from an n-sphere centered at the origin by applying the method of G. MARSAGLIA [12]: For an n-dimensional vector we sample all components $x_1, \ldots, x_n$ from the normal distribution $\mathscr{N}(0, 1)$ with mean zero and variance one. The points are then transformed according to

$$x_j \to x_j' = \frac{r}{\|x\|} x_j + z_j \,, \tag{20.3}$$

where $\|x\|$ denotes the Euclidean norm of the vector x. The points given by Eq. (20.3) lie on the surface of the n-sphere with radius r. In order to obtain uniformly distributed random points *within* a sphere with radius r we draw a random number $u \in [0, 1]$ and calculate

$$x_j \to x_j' = u^{\frac{1}{n}} x_j \,, \tag{20.4}$$

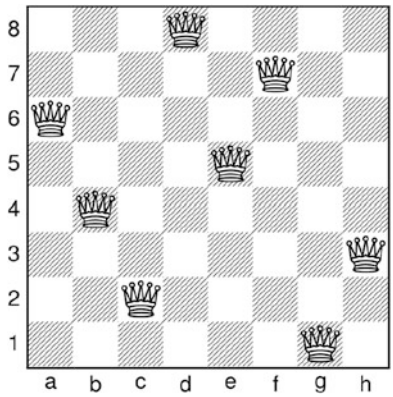
where the factor $1/n$ in the exponent of u ensures that the points are uniformly distributed.

Let us briefly summarize the most important properties of the method of hill climbing:

- The way the algorithm is defined it will terminate in a *local* minimum and not in the global minimum. A classical remedy is the restart of the algorithm from various different initial positions. Information gathered from previous runs can help to make a good choice for the initial positions of restarts.
- It depends highly on the choice of initial conditions if and how the global minimum is found. This situation is very similar to the application of deterministic methods of optimization (see Appendix I). Sometimes it may even be of advantage to accept points which result in a slight increase of the cost function's value just to escape a local minimum.
- For most problems this method is very expensive from a computational point of view.

We apply the method of hill climbing to the N-queens problem for $N = 8$. The algorithm is executed in the following way: In the initial configuration the queens

are set randomly on the chessboard and we place only one queen in each row and column. It is then checked whether or not two queens attack each other. If they do, a new configuration is generated by picking two queens at random and by changing their respective positions. This is repeated until a configuration arises in which none of the queens attacks another. Such an algorithm resembles a random walk in a parameter space which spans all possible configurations under the constraint that only one queen is placed in each row and column. The iteration is terminated as soon as no queen is attacked by any other queen. It is rather obvious that this strategy is not very fast, however, one possible solution to the problem for $N = 8$ can easily be found within a few iteration steps:



However, for large values of N hill climbing is definitely not a recommendable method to solve the N-queens problem.

20.3 Simulated Annealing

Let us turn our attention to *simulated annealing* [7, 13, 14]. The name of this algorithm stems from the annealing process in metallurgy in which a metal is first heated and then slowly cooled in order to reduce the amount of defects in the material. The reasoning behind this method can quite easily be reconstructed with the help of the ISING model which we discussed in detail in Chap. 15. There we learned from thermodynamics that the equilibrium distribution of possible configurations $P(\mathscr{C}, T)$ at a certain temperature T is a BOLTZMANN distribution

$$P(\mathscr{C}, T) = \frac{1}{Z} \exp\left[-\frac{H(\mathscr{C})}{k_{\mathrm{B}} T}\right], \tag{20.5}$$

where $H(\mathscr{C})$ is the HAMILTON function of the system. In particular, we expect that the system is in its ground state (let us assume a non-degenerate ground-state for

the time being) with probability one in the limit $T \to 0$, provided that we cooled sufficiently slowly so that the system had enough time to equilibrate. This can be used to solve the optimization problem: We take the cost function $\mathbb{H}(x)$ and define the probability for the realization of a particular state (configuration) in the search space $x_0 \in \mathbb{S}$ by

$$P(x_0, T) = \frac{1}{Z} \exp\left[-\frac{\mathbb{H}(x_0)}{T}\right], \tag{20.6}$$

where T is some external parameter, which we refer to as *temperature* for reasons of convenience, and Z denotes the normalization constant:

$$Z = \sum_{x \in \mathbb{S}} \mathrm{d}x \exp\left[-\frac{\mathbb{H}(x)}{T}\right]. \tag{20.7}$$

We start the procedure at some finite initial temperature $T_0 \neq 0$ and construct a MARKOV-chain of states $\{x_n\}$ which converges towards the distribution (20.6). We choose, of course, a sampling technique which does not require the explicit knowledge of the normalization Z, such as the METROPOLIS-HASTINGS algorithm of Sect. 18.2. As soon as the MARKOV-chain reaches its stationary distribution for a given temperature T, we slightly decrease the temperature and restart the MARKOV-chain with the last state of the previous temperature. By slowly *cooling* the search MARKOV-chain, we exclude unimportant parts of the search space by decreasing their acceptance probability. Nevertheless, the chain is given enough time to explore the whole remaining search space at each temperature. This procedure is commonly referred to as the classical version of *simulated annealing*.

It is of advantage to start with an initial temperature which allows to cover the largest part of possible states in the search space $\mathbb{S}$. Thus, the acceptance probability for a new state in the MARKOV-chain is almost equal to one for all $x \in \mathbb{S}$. If this were not the case, some regions of the search space might be excluded from our search routine right away due to an unlucky choice of the initial configuration. In particular, the result might be a state in the neighborhood of the initial state of the MARKOV-chain and it is, therefore, most likely a local minimum rather than the global minimum.

We note that the algorithm consists of the following essential ingredients: (i) a proposal probability for new states x within the search space $\mathbb{S}$, (ii) an acceptance probability $P_a(x \to x')$ for a proposed x' from a previous state x, and (iii) a cooling strategy $T = T(t)$, where t is time. Let us briefly elaborate on these points.

(i) Proposal Probability

The question of how to generate new states x from a previous state x' within the search space $\mathbb{S}$ has already been answered in the case of hill climbing, Sect. 20.2 by

defining the neighborhood of a state in search space. The corresponding proposal probability will be denoted by $P_p(x \to x')$.

(ii) Acceptance of Probability

The acceptance probability has to be chosen in such a way that the sequence of generated states constitutes a MARKOV-chain which converges toward the distribution (20.6). Hence, detailed balance has to be imposed and the implications of this requirement have been discussed extensively in Chap. 18. Note that the proposal probability has to be included into the definition of the acceptance probability as was outlined in Sect. 18.2.

One particular choice of a METROPOLIS-HASTINGS acceptance probability

$$P_a(x \to x', T) = \min\left(1, \frac{P(x', T)}{P(x, T)} \frac{P_p(x' \to x)}{P_p(x \to x')}\right) , \tag{20.8}$$

appears to be quite natural for several reasons:

- It is very general and can, thus, also handle asymmetric proposal probabilities.
- In the symmetric case $P_p(x \to x') = P_p(x' \to x)$ and $\mathbb{H}(x') \leq \mathbb{H}(x)$ we get

$$\frac{P(x', T)}{P(x, T)} = \exp\left\{\frac{1}{T}\left[\mathbb{H}(x) - \mathbb{H}(x')\right]\right\} \geq 1 , \tag{20.9}$$

  according to our choice (20.6) and the state x' is accepted with probability one. On the other hand, for $\mathbb{H}(x') > \mathbb{H}(x)$, x' may still be accepted with some finite probability $P_a(x \to x', T)$ which offers an opportunity to escape a local minimum.

(iii) Cooling Strategy

The design of a proper cooling strategy includes both, the choice of an appropriate initial temperature T_0 as well as the formulation of a mathematical rule which defines $T_{n+1} = f(T_n)$ where $T_{n+1} < T_n$.

First of all we discuss the choice of the initial temperature. A common choice is to choose it in such a way that at least 80 % of all generated states are accepted. The simplest procedure to determine this temperature starts with some arbitrary value $T_0 > 0$ and generates N states. If the number of rejected states N_r is greater than $0.2\,N$, then the temperature T_0 is doubled and the number of rejected states is measured again.

Another more sophisticated choice is based on the following idea: The best choice would be $T_0 \to \infty$ because then the acceptance probability would be one for all possible states independent of $\mathbb{H}(x)$. This corresponds to a random walk in search space $\mathbb{S}$ and we calculate the mean value $\langle \mathbb{H} \rangle_\infty$ and the variance $\mathrm{var}\,(\mathbb{H})_\infty$. Thus, the function values $\mathbb{H}$ fluctuate between $[\langle \mathbb{H} \rangle_\infty - \sqrt{\mathrm{var}\,(\mathbb{H})_\infty}, \langle \mathbb{H} \rangle_\infty + \sqrt{\mathrm{var}\,(\mathbb{H})_\infty}]$. We consider now the expectation value $\langle \mathbb{H} \rangle_{T_0}$ for large values of T_0. We define the small parameter $\epsilon = 1/T_0 \ll 1$ and find with $p(x, \epsilon) = P(x, T)$

$$\begin{aligned} \langle \mathbb{H} \rangle_\epsilon &= \int \mathrm{d}x\, p(x, \epsilon) \mathbb{H}(x) \\ &= \langle \mathbb{H} \rangle_0 - \epsilon \left[\langle \mathbb{H}^2 \rangle_0 - \langle \mathbb{H} \rangle_0^2 \right] . \end{aligned} \tag{20.10}$$

Re-substituting $T_0 = 1/\epsilon$ results, finally, in:

$$\langle \mathbb{H} \rangle_{T_0} \approx \langle \mathbb{H} \rangle_\infty - \frac{\mathrm{var}\,(\mathbb{H})_\infty}{T_0} . \tag{20.11}$$

The initial temperature T_0 is now chosen in such a way that the expectation value $\langle \mathbb{H} \rangle_{T_0}$ borders the infinite temperature fluctuations from below and we set consequently

$$\langle \mathbb{H} \rangle_{T_0} = \langle \mathbb{H} \rangle_\infty - \sqrt{\mathrm{var}\,(\mathbb{H})_\infty} , \tag{20.12}$$

with the implication that

$$T_0 = \sqrt{\mathrm{var}\,(\mathbb{H})_\infty} . \tag{20.13}$$

We are now in a position to investigate appropriate cooling strategies: The *geometric cooling schedule*

$$T_n = T_0 q^n , \tag{20.14}$$

with $0 \ll q < 1$ is very often used. However, particular cost functions $\mathbb{H}(x)$ may develop several *phase transitions* in the course of the cooling process. Naturally, the expectation value $\langle \mathbb{H} \rangle$ changes rapidly in the region $T \approx T_c$, with T_c the temperature at which the phase transition occurs. It is, therefore, certainly of advantage to take such a possibility into account and to design the cooling strategy accordingly.

Hence, a more appropriate strategy is to use temperature changes which cause only slightly modified acceptance probabilities. In particular, we demand that

$$\frac{1}{1+\delta} < \frac{P(x, T_n)}{P(x, T_{n+1})} < 1 + \delta , \tag{20.15}$$

with $0 < \delta \ll 1$. Assuming a BOLTZMANN type distribution for $P(x, T_n)$, we obtain

$$\exp\left[-\mathbb{H}(x)\left(\frac{1}{T_n} - \frac{1}{T_{n+1}}\right)\right] < 1 + \delta \,, \tag{20.16}$$

or

$$T_{n+1} > \frac{T_n}{1 + \frac{T_n}{\mathbb{H}(x)} \ln(1+\delta)} \,. \tag{20.17}$$

Hence, we can choose

$$T_{n+1} \approx \frac{T_n}{1 + \frac{T_n}{3\sqrt{\mathrm{var}(\mathbb{H})_{T_n}}} \ln(1+\delta)} \,, \tag{20.18}$$

where we replaced $\mathbb{H}(x) \approx 3\sqrt{\mathrm{var}\,(\mathbb{H})_{T_n}}$. This choice is plausible if one recognizes that we can replace $\mathbb{H}(x) \to \mathbb{H}(x) - \mathbb{H}_{\min}$ in the above calculations, where $\mathbb{H}_{\min}$ represents the (unknown) minimum of $\mathbb{H}(x)$. This cooling schedule is known as the AARTS *schedule*.

Finally, we have to discuss how to terminate the algorithm. Typically, there are several choices and we present briefly the most popular ones. The obvious choice is to terminate the algorithm as soon as the acceptance ratio is below some predefined threshold value. A more sophisticated choice is to terminate the algorithm whenever the mean value $\langle \mathbb{H} \rangle$ reaches some constant value. A quite different and more formal approach would be to initially define a maximum number of iterations or to set the final temperature T_f to some reasonable value. Nevertheless, the termination condition has to be defined for each particular problem individually.

Before presenting an example, we note some further results associated with cooling strategies. It was demonstrated by S. KIRKPATRIK et al. [15] that the optimal cooling strategy for a BOLTZMANN type distribution is of the form

$$T_n \propto \frac{1}{\ln(n)} \,, \tag{20.19}$$

where n labels the temperature steps. In this case the global minimum is found with probability one. However, the convergence is rather slow. In addition, several extensions of classical simulated annealing have been suggested in the literature. For instance, *fast simulated annealing* uses a CAUCHY distribution

$$P(x, T) = \frac{T}{(x^2 + T^2)^{\frac{d+1}{2}}} \tag{20.20}$$

instead of a BOLTZMANN distribution. Here d is the dimension of the search space $\mathbb{S}$. The optimal cooling strategy for such a distribution function is of the form

$$T_n \propto \frac{1}{n} \,, \tag{20.21}$$

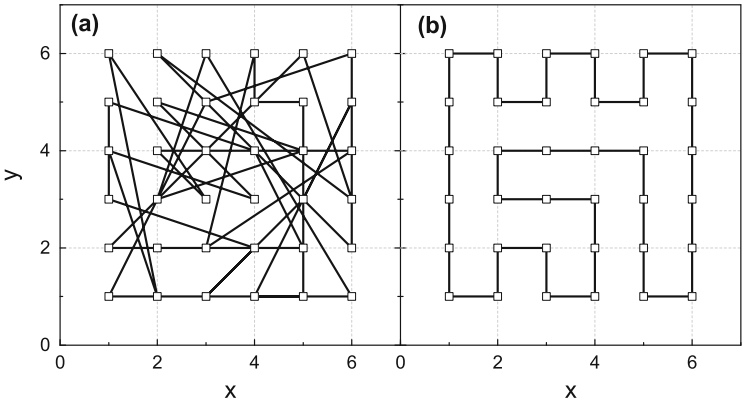


Fig. 20.1 **(a)** Initial route of the traveling salesperson for 36 cities on a regular grid. **(b)** One of many optimal routes of the traveling salesperson for 36 cities on a regular grid

which signifies a considerable increase in convergence speed in comparison to Eq. (20.19). Another generalization is referred to as *generalized simulated annealing* and is based on the TSALLIS distribution which depends on an external parameter ϵ:

$$P_\epsilon(x,T) = \frac{1}{Z}\left[1 + \frac{\epsilon \mathbb{H}(x)}{k_\mathrm{B}T}\right]^{-\frac{1}{\epsilon}} . \tag{20.22}$$

It can be demonstrated that P_ϵ converges toward the BOLTZMANN distribution for $\epsilon \to 0$. We mention in passing that the concept of the TSALLIS distribution is closely intertwined with the definition of the TSALLIS entropy and the formulation of non-extensive thermodynamics by C. TSALLIS [16].

As a first illustrative example we discuss the *traveling salesperson problem* for $N = 36$ cities on a regular grid because in this case the optimal route is easily identified. We calculate the initial temperature from Eq. (20.13) and employ the geometric cooling schedule (20.14) with $q = 0.99$ together with a termination criterion of the form

$$\langle \mathbb{H} \rangle_{T_n} - \langle \mathbb{H} \rangle_{T_{n-1}} < \eta \, , \tag{20.23}$$

where η is the required accuracy. Figure 20.1a presents one route for the initial temperature and Fig. 20.1b displays one of many optimal routes after convergence has been reached. This case will be called the *first scenario*. In the *second scenario* we place 36 cities in four equally spaced clusters. Results for the optimal route are presented in Fig. 20.2b.

The possibility of phase transitions to occur during the cooling process has already been mentioned. In a genuine physical system the question whether a phase transition is possible at all or if it is of first or second order is solely determined by

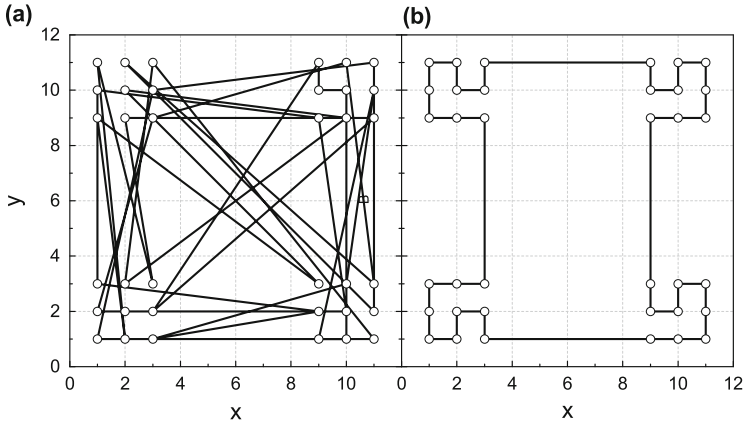


Fig. 20.2 **(a)** Initial route of the traveling salesperson for 36 cities placed in four equally spaced clusters. **(b)** One of many optimal routes of the traveling salesperson for 36 cities placed in four equally spaced clusters

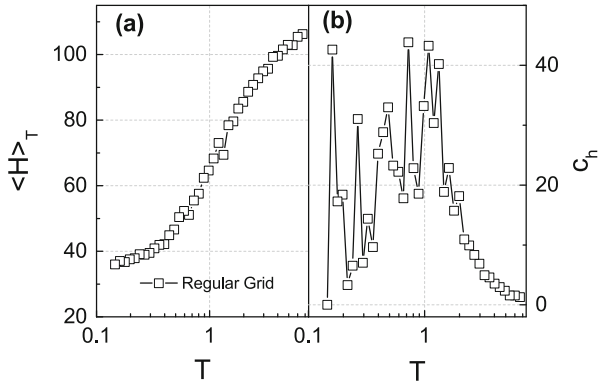


Fig. 20.3 **(a)** The expectation value $\langle \mathbb{H} \rangle_T$ and **(b)** the 'specific heat' c_h vs temperature T for scenario one

the HAMILTON function $H(x)$ of the system. As an intriguing example we refer to the q-states POTTS model of Sect. 18.3 where a second order phase transition was observed for $q \leq 4$ and a first order phase transition for $q > 4$. In analogy, the order of a 'phase transition' during the iteration process toward the global minimum in simulated annealing is completely determined by the particular properties of the cost function $\mathbb{H}(x)$. We want to study such a possibility and determine the expectation values $\langle \mathbb{H} \rangle_T$ and the 'specific heat' c_h as functions of temperature T for the two scenarios of the traveling salesperson problem. Figure 20.3 presents the results for scenario one and Fig. 20.4 those for scenario two. The second scenario develops two second order phase transitions while in the first scenario only one second order phase transition can be observed. The first phase transition of the second scenario

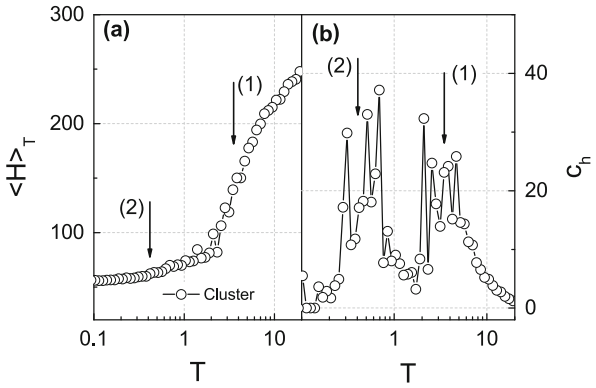


Fig. 20.4 The same as Fig. 20.3 but for scenario two. Two second order phase transitions are observed. They are indicated by *down arrows* labeled *(1)* and *(2)*

(at $T \approx 3.5$) can be related to the optimization of the clusters' sequence while in the second phase transition (at $T \approx 0.42$) the sequence of cities within the clusters becomes finalized. These two transitions are indicated by down arrows labeled (1) and (2) in Fig. 20.4.

20.4 Genetic Algorithms

The sparkling idea of *genetic algorithms* has originally been lent from natures *survival of the fittest* [17]. The basic intentions are quickly summarized by remembering the natural evolution of a particular species within a hostile environment: The individuals of the species *reproduce* from one *generation* to another. During this process the *genes* of the individuals are modified by local *mutations*. Individuals best accustomed to the environment then survive with higher probability. This very last process is referred to as *selection*. By iterating this process for large *populations* the individuals of the whole species will adjust their properties to the environment *on average*,[1] and, thus, the individuals will be better equipped for survival within the hostile environment. A large population is compulsory in order to obtain a huge variety in the *phenotype* of the individuals. Algorithms based on such a scheme are referred to as *genetic algorithms*.

We are not going into the details of the implementation of genetic algorithms because this is beyond the scope of this book. However, the ideas sketched above will be applied to the problem of the traveling salesperson passing through m-cities just to illustrate the method. Let $s = (s_1, \ldots, s_m) \in \mathbb{N}^m$ denote a list of m integers, which obey $s_i \leq i$. For instance, for $m = 10$, s might be given by

[1] Note that in the real world the environment (in particular the natural enemies of a species) develop as well. Moreover, we do not consider any communication within a species, like the formation of *societies*, *learning*, and related processes.

Table 20.1 Sample tour to illustrate the recovery of the order of cities within a genetic algorithm. Elements indicated by [x] are 'selected' elements which are added to the column Tour

$\hat{s}$	1	2	3	4	5	6	7	8	9	10		Tour
9	1	2	3	4	5	6	7	8	[9]	10	→	9
4	1	2	3	[4]	5	6	7	8	10		→	4
3	1	2	[3]	5	6	7	8	10			→	3
3	1	2	[5]	6	7	8	10				→	5
5	1	2	6	7	[8]	10					→	8
1	[1]	2	6	7	10						→	1
4	2	6	7	[10]							→	10
2	2	[6]	7								→	6
2	2	[7]									→	7
1	[2]										→	2

$$s = (1, 2, 2, 4, 1, 5, 3, 3, 4, 9)\ . \tag{20.24}$$

The order of cities is then recovered by setting $\hat{s} = (s_m, \ldots, s_1)$ and performing the steps illustrated in Table 20.1.

In words: The vector $\hat{s}$ labels the elements taken from the list $(1, 2, \ldots, m)$ *with* removal. The resulting list *Tour* specifies the optimum sequence of the cities. The genetic algorithm is executed in the following steps:

- Define M initial individuals.
- *Mutation*: for each individual we introduce a single random local modification with probability p_{mut}.
- *Reproduction*: We produce M additional individuals by pairwise combining the *parents*. This is performed by

  (a) Pick two individuals at random.
  (b) Draw a random integer $r \in [1, m-1]$ and replace the first r genes of the first individual by the first r genes of the second individual and vice versa.

  In this way, we obtain $2M$ individuals.
- *Selection*: The M individuals with the highest *fitness* which corresponds to the lowest value of the cost function survive.

The above steps are repeated until the desired number of generations has been achieved.

In Fig. 20.5 we show the optimal path for the traveling salesperson problem discussed in the previous section, but now for $N = 30$ cities. It was obtained with the genetic algorithm described here. The number of individuals was chosen to be $M = 5000$ and the number of generations to be $G = 5000$.

Some remarks are appropriate: First of all we note that there are many different permutations of how a genetic algorithm can be realized. In particular, it is the problem which determines the most convenient form to implement the essential

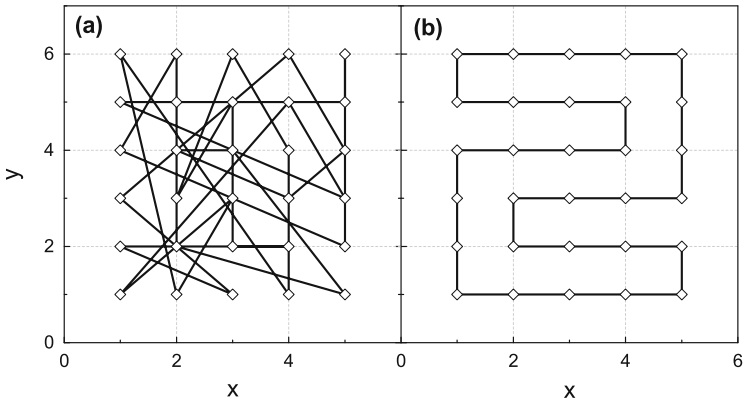


Fig. 20.5 **(a)** The random route of one individual out of the population of 5000. **(b)** One of many optimal routes of the traveling salesperson for $N = 30$ cities as obtained by a genetic algorithm

ingredients: mutation, reproduction, and selection. However, particular care is required in formulating the algorithm in such a way that it does not produce individuals which are *too* similar. In such a case the algorithm is very likely to terminate in a local minimum.

Another remark comments on how to treat optimization problems with continuous variables x. Here it might be advantageous to represent the variable x in its binary form because it makes the reproduction step particularly simple.

20.5 Some Further Methods

We briefly list some alternative stochastic optimization techniques without going into detail. Two famous alternatives which are closely related to simulated annealing are:

- *Threshold Accepting Algorithms*: The new configuration x' is accepted with probability one if $\mathbb{H}(x') \leq \mathbb{H}(x) + T$. During the simulation the *temperature* or *threshold level* T is continuously decreased. The above choice of an acceptance probability is very effective to allow for an escape from local minima.
- *Deluge Algorithms*: These algorithms are very similar to threshold accepting algorithms. We present it in the original formulation which is suited to find the global maximum of a function $\mathbb{G}(x)$. The global minimum of $\mathbb{H}(x)$ can be found by searching the maximum of $\mathbb{G}(x) = -\mathbb{H}(x)$. One accepts a new state x' with probability one if $\mathbb{G}(x') > T$, where T is continuously increased during the simulation. Hence, the whole *landscape* of $\mathbb{G}(x)$ is *flooded* with increasing T until only the summits of $\mathbb{G}(x)$ are left. Finally, only the biggest *mountain* will reach out of the *water* and the global maximum has been found.

Two famous ideas which are closely related to genetic algorithms are:

- *Grouping Genetic Algorithms*: The idea is to put the individuals of the population in distinct *groups*. These groups may for instance be formed by comparing the genes or grouping individuals with similar cost function values. All members of a group have one part of the genes in common, and all operators acting on genes act on the whole group. Such an approach can significantly improve the convergence rate of a classical genetic algorithm.
- *Ant Colony Optimization*: The idea is, again, borrowed from nature, in particular from an ant colony searching the optimum path between two or more fixed or variable points. In a real world an ant travels from one point to another randomly, leaving a trail of pheromone on its traveled path. Following ants are very likely to follow the pheromone trail, however, some random nature remains. The key point is that with time the pheromone trail starts to evaporate, hence its impact on the path of following ants is reduced if the path is not traveled frequently or often enough so that the pheromones evaporated. In this way one prevents the algorithm to get stuck in a local minimum and the global minimum may be found by sending out artificial ants.

There are many further methods available in the literature (see, for instance, Refs. [2, 18]) to which we refer the interested reader.

Summary

The local maximum/minimum of some cost function $\mathbb{H}(x)$ within a search space $\mathbb{S}$ can be determined using stochastic methods, thus establishing a particular class of algorithms known as *Stochastic Optimization*. The most straightforward method was the algorithm of *hill climbing* which resembled a controlled random walk within a restricted search space $\mathbb{S}$ called neighborhood. Because of this feature hill climbing will find in general local minima within this neighborhood and the global minimum has to be found under variation of initial conditions. This made this method too expensive for more complex problems from a computational point of view. To move from a random walk formulation to a formulation on the basis of MARKOV-chain Monte Carlo was the logical next step. The method of choice was named *simulated annealing*. It used the METROPOLIS-HASTINGS algorithm to generate new configurations within a search space $\mathbb{S}$ from a temperature dependent equilibrium distribution. A cooling strategy was used to slowly restrict the search space to the neighborhood of the global minimum. This global minimum was always found, albeit rather slowly. We mentioned some flavors of this basic algorithm which either differed in the definition of the acceptance probability or in the cooling strategy. A completely different class of algorithms was established with the so-called *genetic algorithms*. They were adapted from nature's concept of the survival of the fittest. They were based on the notions of: (i) Mutation, a single random local modification of a certain probability. (ii) Reproduction, additional 'individuals' were

generated by pairwise combining parents. (iii) Selection: Individuals with the lowest value of the cost function survived and mutation started again. Genetic algorithms established a very versatile class of solvers to cover a huge body of optimization problems.

Problems

Solve the traveling salesperson problem for $N = 20$ cities on a regular grid with the help of simulated annealing. As a cooling schedule, use the geometric cooling as explained in Sect. 20.3. Determine the initial temperature by demanding an acceptance rate of 90 % and terminate the algorithm if the mean value of the cost function $\langle \mathbb{H} \rangle$ remains unchanged for at least 10 successive temperatures. Calculate the expectation value $\langle \mathbb{H} \rangle_T$ for different temperatures and identify the transition temperature. In a second step produce a list of 20 cities which are randomly distributed on a two-dimensional grid. Optimize this problem as well. Note that you should produce the list of cities only *once* in order to obtain comparable and reproducible results.

References

1. Panos, M.P., Resende, M.G.C. (eds.): Handbook of Applied Optimization. Oxford University Press, New York (2002)
2. Hartmann, A.H., Rieger, H.: Optimization Algorithms in Physics. Wiley – VCH, Berlin (2002)
3. Locatelli, M., Schoen, F.: Global Optimization. MPS-SIAM Series on Optimization. Cambridge University Press, Cambridge (2013)
4. Scholz, D.: Deterministic Global Optimization. Springer, Berlin/Heidelberg (2012)
5. Schneider, J.J., Kirkpatrick, S. (eds.): Stochastic Optimization. Springer, Berlin/Heidelberg (2006)
6. Lawler, E.L., Lenstra, K.K., Rinnooy Kan, A.H.G., Shmoys, D.B.: The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization. Wiley, New York (1985)
7. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C++, 2nd edn. Cambridge University Press, Cambridge (2002)
8. Kundu, S., Acharyya, S.: Stochastic local search approaches in solving the nurse scheduling problem. In: Chaki, N., Cortesi, A. (eds.) Computer Information Systems – Analysis and Technologies, Communications in Computer and Information Science, vol. 245, pp. 202–211. Springer, Berlin/Heidelberg (2011)
9. Marinari, E., Parisi, G., Ritort, F.: On the 3d Ising spin glass. J. Phys. A: Math. Gen. **27**, 2687 (1994). doi:10.1088/0305-4470/27/8/008
10. Watkins, J.J.: Across the Board: The Mathematics of Chessboard Problems. Princeton University Press, Princeton (2012)
11. Skiena, S.S.: The Algorithm Design Manual. Springer, Berlin/Heidelberg (2008)
12. Marsaglia, G.: Choosing a point from the surface of a sphere. Ann. Math. Stat. **43**, 645–646 (1972). doi:10.1214/aoms/1177692644
13. Bertsimas, D., Tsitsiklis, J.: Simulated annealing. Stat. Sci. **8**, 10–15 (1993)

14. Salamon, P., Sibani, P., Frost, R.: Facts, Conjectures, and Improvements for Simulated Annealing. Cambridge University Press, Cambridge (2002)
15. Kirkpatrik, S., Gellat, C.D., Jr., Vecchi, M.P.: Simulated annealing. Science **220**, 671 (1983)
16. Tsallis, C.: Introduction to Nonextensive Statistical Mechanics. Springer, Berlin/Heidelberg (2009)
17. Man, K.F., Tang, K.S., Kwong, S.: Genetic Algorithms. Springer, Berlin/Heidelberg (1999)
18. Bäck, T.: Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York (1996)